

HƯỚNG DẪN CÀI ĐẶT ROS

MỤC LỤC

1.	Khái niệm.....	1
2.	Các đơn vị của ROS.....	1
3.	Cấu trúc thư mục của ROS.....	3
4.	Cài đặt ROS.....	4
5.	Tạo workspace trên ROS:	5
6.	Các công cụ KHÁC.....	6

1. KHÁI NIỆM

Robot operating system (ROS) là một hệ thống phần mềm chuyên dụng để lập trình và điều khiển robot, bao gồm các công cụ để lập trình, hiển thị, tương tác trực tiếp với phần cứng, và kết nối cộng đồng robot trên toàn thế giới. Nói chung là nếu bạn muốn lập trình và điều khiển một con robot, sử dụng phần mềm ROS sẽ giúp quá trình thực hiện nhanh hơn và đơn giản hơn rất nhiều. Và bạn không cần ngồi viết lại những thứ mà người khác đã làm rồi, mà có những thứ muốn viết lại cũng không có khả năng.

Về cơ bản, ROS là 1 nền tảng để giao tiếp giữa 2 chương trình hay bộ xử lý. Ví dụ, ta có 2 đơn vị xử lý A và B chạy độc lập, nếu A muốn gửi data cho B hoặc ngược lại thì ta có thể thực hiện nhờ vào ROS. Ta có thể thực hiện được nhiều đơn vị xử lý trong ROS chạy độc lập với nhau và các đơn vị xử lý có thể giao tiếp với nhau. Đây là lý do ta chọn lựa ROS để thực hiện.

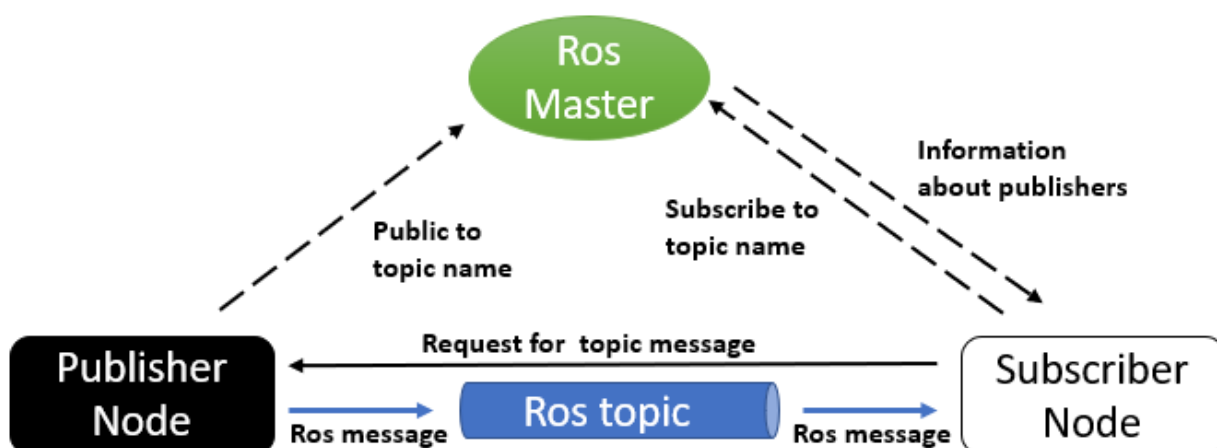
2. CÁC ĐƠN VỊ CỦA ROS

Về cơ bản ROS files được bố trí và hoạt động như sau, từ trên xuống theo thứ tự như sau, metapackages, packages, packages manifest, Misc, messages, Services, codes:

- Trong đó gói tổng(Metapackages) là một nhóm các gói(packages) có liên quan tới nhau.
- Gói (Packages): chính là các nguyên tử cơ bản nhất tạo nên ROS. Trong một gói gồm có, ROSnode, datasets, configuration files, source files, tất cả được gói lại trong một "gói". Chúng ta chỉ cần quan tâm 2 thứ trong 1 gói, đó là src folder - chứa source code của chúng ta, và file Cmake.txt - khai báo những thư viện cần thiết để thực thi code
- Nodes: các đơn vị tính toán độc lập của ROS. Một robot có thể có rất nhiều nodes để thực hiện quá trình giao tiếp của nó. Ví dụ như, một con robot tự lái sẽ có những node sau, node đọc dữ liệu từ Laser scanner, Kinect camera, localization and mapping, node gửi lệnh vận tốc cho hệ thống bánh lái. Khi chạy chương trình, các node gửi tất cả thông tin của nó lên ROS master bao gồm kiểu dữ liệu mà node đó gửi đi hay nhận được. Node mà gửi data đi được gọi là publisher và node nhận data gọi là subscriber.
- Master: ROS master đóng vai trò như một node trung gian kết nối giữa các node khác nhau. Master bao quát thông tin về tất cả các nút chạy trong môi trường ROS. Nó sẽ trao đổi chi tiết của một nút với khác để thiết lập kết nối

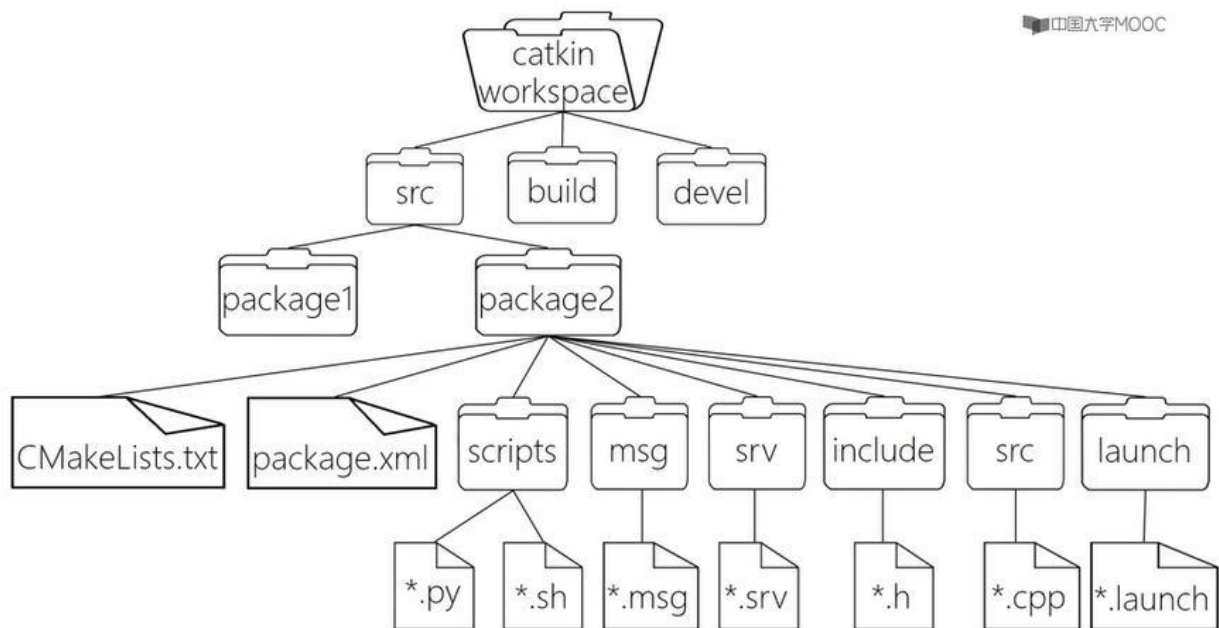
giữa chúng. Sau khi trao đổi thông tin, giao tiếp sẽ bắt đầu giữa hai nút ROS. Nếu không có Master, các nút sẽ không thể tìm thấy nhau, trao đổi thông điệp hoặc gọi các dịch vụ.

- **Topic:** Một trong những phương pháp để giao tiếp và trao đổi tin nhắn giữa hai node được gọi là ROS Topic. ROS Topic giống như một kênh tin nhắn, trong kênh đó dữ liệu được trao đổi bằng ROS message. Mỗi chủ đề sẽ có một tên khác nhau tùy thuộc vào những thông tin mà nó sẽ phụ trách cung cấp. Một Node sẽ publish thông tin cho một Topic và một nút khác có thể đọc từ Topic bằng cách subscribe với nó.
- **Service:** một loại phương pháp giao tiếp khác với Topic. Topic sử dụng tương tác publish hoặc subscribe nhưng trong service, nó tương tác theo kiểu request – response. Server đặc biệt hữu dụng khi chúng ta cần thực hiện một lệnh gì đó mà cần thời gian lâu để xử lý, vậy thì chúng ta để nó ở server, khi cần thì ta gọi



Hình 2-1 Kết nối giữa các node và topic trong ROS

3. CẤU TRÚC THƯ MỤC CỦA ROS



Hình 3-1 Cấu trúc thư mục workspace

- **Workspace:** Nơi chứa project của ROS. Hai folder build và devel tự tạo ra khi chúng ta build chương trình. Thư mục src chứa code người dùng.
- **Packages:** Mỗi packages sẽ có các nhiệm vụ riêng tùy thuộc vào từng project. Tất cả source code, data files, build files, dependences, và các file khác đều được chứa trong các packages.
- Trong mỗi packages sẽ chứa các node và thư viện sử dụng cho các node đó: A package manifest: là 1 file .xml được đặt bên trong mỗi package, nó có tất cả thông tin cơ bản của 1 package, bao gồm tên package, mô tả, tên tác giả, các gói mà package đó phụ thuộc,...

4. CÀI ĐẶT ROS

Vào trình duyệt Ubuntu. Truy cập <http://wiki.ros.org/ROS/Installation>

ROS có 2 phiên bản là **kinetic** và **melodic** → chọn phiên bản nào cũng được nhưng phải nhớ phiên bản cài đặt.

Mở terminal trên Ubuntu Lần lượt cài đặt các dòng lệnh sau (chú ý: nếu dùng phiên bản kinetic thì sửa **melodic** thành **kinetic**)

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >
/etc/apt/sources.list.d/ros-latest.list'

sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

sudo apt update

sudo apt install ros-melodic-desktop-full

apt search ros-melodic

echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc

source ~/.bashrc

sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool
build-essential

sudo apt install python-rosdep

sudo rosdep init

rosdep update
```

5. TẠO WORKSPACE TRÊN ROS:

Bước đầu trong việc xây dựng một application trên nền tảng ROS, ta cần workspace. Nơi này sẽ dùng để chứa các package cũng như build chúng. Trước tiên ta cần tạo 1 thư mục workspace:

<code>mkdir catkin_ws</code>	<code>// tạo thư mục catkin_ws</code>
<code>cd catkin_ws</code>	<code>// chuyển đến thư mục catkin_ws</code>
<code>mkdir src</code>	<code>// tạo thư mục src là con của catkin_ws</code>
<code>cd src</code>	<code>// chuyển đến thư mục src</code>
<code>catkin_init_workspace</code>	<code>// khởi tạo ws</code>
<code>cd ..</code>	<code>// chuyển đến ws</code>
<code>catkin_make</code>	

Ta có thể thấy là có thêm 2 thư mục build và devel

- Thư mục src: là nơi chứa các package, ta có thể thêm 1 package có sẵn bên ngoài vào đây.
- Thư mục build: khi build sẽ xuất hiện 1 thư mục mang tên package đó trong này và kèm thêm 1 số file khác.
- Thư mục devel: có một thư mục include. Ngoài ra, sẽ có 1 file setup.bash, đây là file cần được kích hoạt khi ta muốn chạy bất kì package nào trong ws(workspace)

6. CÁC CÔNG CỤ KHÁC

Rviz: công cụ hiển thị ba chiều được sử dụng để hình dung robot, môi trường chúng làm việc và dữ liệu cảm biến. Nó là một công cụ có cấu hình cao, với nhiều loại trực quan hóa và plugin khác nhau.

Rosbag: sử dụng để ghi và phát lại dữ liệu tin nhắn ROS. Rosbag sử dụng một định dạng tệp được gọi là túi (bag), ghi lại các thông báo ROS bằng cách nghe các chủ đề và ghi lại các thông điệp khi chúng được gửi đến. Việc phát lại các thông báo từ một túi phần lớn giống như việc có các nút ban đầu tạo ra dữ liệu trong tính toán ROS đồ thị, làm cho túi trở thành một công cụ hữu ích để ghi dữ liệu để sử dụng trong quá trình phát triển sau này. Trong khi rosbag là công cụ chỉ dòng lệnh, rqt_bag cung cấp giao diện GUI cho rosbag.

Catkin: hệ thống xây dựng ROS, thay thế cho rosbuidl. Catkin dựa trên Cmake, đa nền tảng, không phụ thuộc vào ngôn ngữ (C, Python, ...)

Roslaunch: sử dụng khi cần chạy nhiều node hay thiết lập các tham số. Một project sẽ có thể có nhiều file launch tùy theo các ứng dụng của người dùng. Trong quá trình chạy chương trình chỉ chạy được 1 file launch duy nhất.